Shape-Aware Material: Interactive Fabrication with ShapeMe

Michael Wessely
michael.wessely@inria.fr

Theophanis Tsandilas
theophanis.tsandilas@inria.fr

Wendy E. Mackay
wendy.mackay@inria.fr

Inria; Univ. Paris-Sud & CNRS (LRI); Université Paris-Saclay
F-91405 Orsay, France

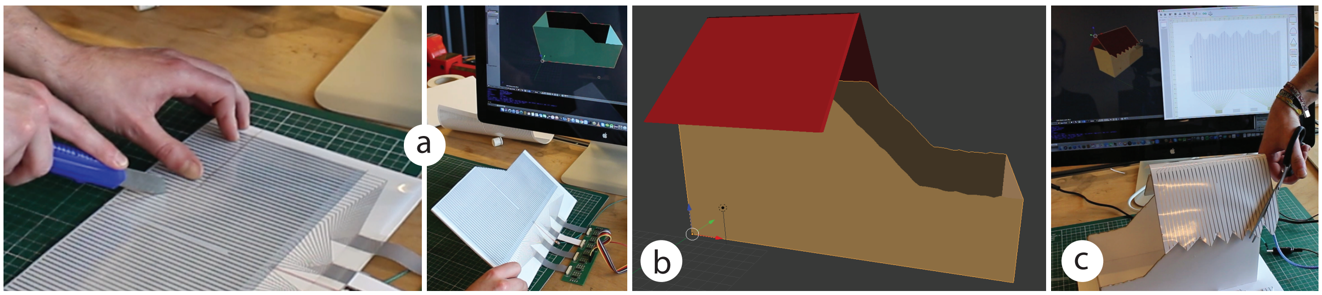

Figure 1. ***ShapeMe*** **is a novel sensing technology that enables physical modeling with shape-aware material: (a) The maker cuts a foamcore piece to reshape the walls of a house model. The updated shape is captured by a grid of length-aware sensors and is communicated to 3D modeling software. (b) The makers digitally creates the pieces of the roof and then produces its physical model. (c) The maker explores variations of the roof by cutting its side with scissors, while its shape is continuously captured.**

ABSTRACT
Makers often create both physical and digital prototypes to explore a design, taking advantage of the subtle feel of physical materials and the precision and power of digital models. We introduce *ShapeMe*, a novel smart material that captures its own geometry as it is physically cut by an artist or designer. *ShapeMe* includes a software toolkit that lets its users generate customized, embeddable sensors that can accommodate various object shapes. As the designer works on a physical prototype, the toolkit streams the artist's physical changes to its digital counterpart in a 3D CAD environment. We use a rapid, inexpensive and simple-to-manufacture inkjet printing technique to create embedded sensors. We successfully created a linear predictive model of the sensors' lengths, and our empirical tests of *ShapeMe* show an average accuracy of 2 to 3 mm. We present an application scenario for modeling multi-object constructions, such as architectural models, and 3D models consisting of multiple layers stacked one on top of each other. *ShapeMe* demonstrates a novel technique for integrating digital and physical modeling, and suggests new possibilities for creating shape-aware materials.

Michael Wessely, Theophanis Tsandilas, and Wendy E. Mackay. Shape-Aware Material: Interactive Fabrication with ShapeMe. In Proceedings of the 31st ACM Symposium on User Interface Software and Technology (UIST), ACM, October 2018.

DOI: https://doi.org/10.1145/3242587.3242619

Author Keywords
Personal fabrication, sensing technologies, shape-aware material, printed electronics, physical modeling.

INTRODUCTION
Industrial designers, architects, sculpteurs, and other makers commonly create multiple physical prototypes to explore alternative shapes. The traditional prototyping process involves creating a series of physical models. For example, architects prototype miniature buildings in cardboard, foamcore, or other materials to get a stereoscopic view of their relative dimensions, and how they look under different lighting conditions [38]. This rapid *cut-view-cut-view* process helps provide a holistic impression of the design and may inspire new ideas. Unfortunately, creating slight (or major) variations is time-consuming and sometimes frustrating: once cut, the new shape of the physical prototype no longer corresponds to the digital version, and may be expensive, difficult or impossible to recapture precisely, especially with soft materials.

Some makers learn sophisticated digital 3D modeling tools so they can directly laser cut or 3D print intermediate prototypes. Unfortunately, CAD software is notoriously difficult to learn and is not designed for rapid prototyping. Given the trade-offs, makers often alternate between physical and digital models, taking advantage of the rapid, tactile interaction with physical models of various sizes, and the precision and transformative capabilities of their digital counterparts. The current process for generating a physical model from a digital one is relatively straightforward, given the right laser cutter or 3D printer. Moving from a physical to a digital model is more difficult. Makers can either precisely measure the physical prototype or use calibrated cameras to capture a digital model of their prototype in a CAD system.

We argue that makers need *shape-aware* prototyping materials that record the results of each cut in digital form. This allows makers to easily transition between physical and digital models during the early prototyping stage of a design. We present *ShapeMe*, a self-aware material that can be cut by the artist with manual tools, and continuously update the current shape of a corresponding digital model. The artist can also use software tools to modify the digital model and reproduce it in physical form. We show how shape-aware physical models can be produced with techniques of printed electronics [10, 17].

ShapeMe uses a lossy capacitor system to identify the length of thin line-shaped sensors. We show how to extract the shape of a 2D or 3D object from data captured from grids of these sensors. The *ShapeMe* toolkit lets users take advantage of where they expect to cut to optimize the design of their sensors. We demonstrate how ink-jet printing or silk-screen printing offers a simple, rapid and inexpensive fabrication technique for embedding shape-aware sensors directly on to various materials. Unlike external scanning approaches [36, 21] such as depth- or stereo cameras, *ShapeMe* does not suffer from occlusion and needs no external systems.

In summary, our core contributions include:

1. *ShapeMe*, a novel sensing technology that detects the shape of objects while being cut. It can be embedded into a variety of prototyping materials and can support both 2D and 3D physical models.
2. A model to estimate the length of *ShapeMe* sensors. Our technical evaluation quantifies the accuracy of this model.
3. The *ShapeMe* software toolkit that lets users design their *ShapeMe* models and link them to 3D modeling software.

We illustrate novel uses of our technology with a scenario walkthrough, inspired by architectural modeling.

RELATED WORK

Our work builds upon research in interactive fabrication, programmable matter, and printed electronics.

Interactive Fabrication

The term *Interactive Fabrication*, first introduced by Willis et al. [41], describes a design process where makers get tangible feedback while working with digital models. Previous HCI research introduced diverse approaches to support this process.

One research area uses external scanning devices to capture the shape of each physical model and assist in the digital modeling process. For example, CopyCAD [4] enables modelers to interactively fabricate a remixed model with a milling machine by capturing the shape of the object with a 2.5D scanner. Along the same lines, ReForm [36] combines a 3D scanner with a 3D printer, which supports manually shaping a clay model before editing it via digital-modeling operations. RealFusion [21] digitizes physical objects with a depth camera. Modelers then execute mid-air gestures that modify the digital geometry. Savage et al. [29] further annotate physical models with stickers, after which a digital tool extracts the stickers' image from the scan to integrate functional components, such as mechanical parts or electronic components.

Another research area focuses on supporting live interaction with fabrication machines as they print a model. Constructables [14] use laser pointers as controllers to construct objects with a laser cutter, from scratch. D-Coil [20] supports digital modeling with an actuated wax extruder to create tangible shape proxies. On-the-fly Print [19] extends this work by allowing modelers to observe a continuously updated physical model while working on 3D CAD software. In a later iteration, Peng et al.[18] presented an augmented-reality system that lets designers create and interact with a virtual model while a robotic arm prints the actual physical model.

Other researchers create guidance systems that support the manual modeling process itself. Rivers et al. [26] use a camera-projector pair to support manual sculpting by showing visual cues on the surface of each object. Zoran et al. [44] introduce a hand-held milling device that physically constrains the crafting process based on a digital model, while preserving the modeler's freedom to manipulate the work in creative ways. WireDraw [43] uses mixed-reality guided drawing to support construction with a 3D extruder pen.

A final approach relies on physical proxies to enhance digital modeling with tangible feedback, or uses specialized modeling devices that can sense the physical interactions of their users. For example, ModelCraft [31] lets architects edit their 3D models by physically annotating paper prototypes printed with an Anoto pattern. DressUp [40] enables fashion designers to create a digital model of a dress by working on a physical mannequin. Spata [35] provides a tangible measurement tool to add real object constraints while designing digitally. Finally, StrutModeling [12] is a construction kit for producing physical models that sense their composition and generating a digital model of themselves.

ShapeMe's approach belongs to this last category, but supports a different type of physical model and manipulation. Unlike ModelCraft [31], which focuses on similar types of prototyping material but only captures pen annotations, *ShapeMe* senses the real shape of the printed physical model.

Shape-Changing Displays and Programmable Matter

Other research related to shape-aware objects includes *shape-changing displays* and *programmable matter*. Shape displays are physical surfaces or volumes that can sense user input and provide a computer-controlled geometry [23, 28]. Several systems offer 2.5D control of the surface via controllable bars [22, 13] or pneumatic actuation [5]. Unfortunately, these approaches require specialized equipment and can only detect a limited range of shapes.

The concept of programmable matter, inspired by Sutherland [32] and Ishii [9], has also been explored by robotics researchers. Self-configurable robotic cubes [27, 34] take programmed 3D shapes and, when moved manually, sense their current configuration. Other systems rely on swarm robots that can form 2D-shapes on diverse surfaces [11] or cloth [3]. Implementation of these systems relies on large, complex robotic units, making them unsuitable for modeling scenarios requiring inexpensive and easily available prototyping materials.

Fabrication of 2D Interfaces with Printed Electronics

Print electronics offer the key enabling technology for the *ShapeMe* approach, offering low-cost fabrication of thin and flexible sensors that can cover large areas. Researchers have demonstrated how to rapidly fabricate flexible interfaces using a desktop ink-jet printer [10], screen printing [17], and water-transfer printing [7]. Others have developed touch sensing [6, 24], deformation sensing [25, 33], stretchable interfaces [37, 39], and shape-changing interfaces [8]. More recently, Oh et al. [15] showed how to create interactive 3D objects by stacking together many layers of conductive paper. Olberding et al. [16] also introduced a touch sensor that is robust to cutting and can be reshaped after fabrication with manual tools. *ShapeMe* builds upon this previous research to offer the first geometry-sensing technology based on print electronics.

EARLY STUDIES WITH EXPERT AND NOVICE MAKERS

The *ShapeMe* approach and scenarios are inspired from early studies with professional designers and artists as well as the results of an informal prototyping workshop. We summarize key lessons from these studies below.

Interviews with Expert Practitioners

We interviewed a chief architect and two modelers from two large architectural firms with extensive experience in physical modeling methods. The modelers showed us a variety of physical models that combine both digitally fabricated parts but also parts created manually with traditional cutting tools. The chief architect explained that physical mockups used to play an important role for communicating with external stakeholders, such as clients and competition juries, but have been largely replaced by computer-based presentation tools. Even so, they still use physical modeling to iterate on design ideas. These iterations involve physical prototypes at multiple scales and often require close collaboration between the architect and modeler. The architect said he often needs to edit physical models by directly removing pieces with a cutter knife.

We also interviewed a shoemaker who works in a company that produces shoes for people with special needs. His design process combines digital and physical modeling tools, of which the most fundamental component involves shaping the sole. Each sole consists of three to five physical layers that are stacked together to create a thicker 3D structure. The shoemaker explained that these layers were physically modeled and then cut, one by one. The primary challenge is how to capture the cuts from each layer so as to produce model templates for the other layers of the sole. The process is highly time consuming, with multiple manual steps involved in measuring the cut material.

ShapeMe's fabrication workflow is highly inspired by professional practices of architects and shoemakers, where a 3D physical model is divided into multiple 2D layers or planes. This layering approach is a key part of our sensing design.

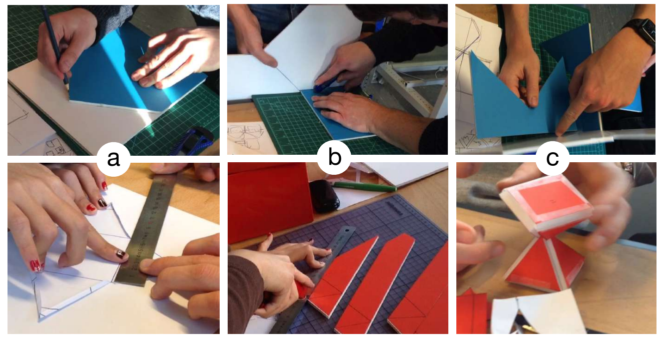

Figure 2. Representative strategies of the blue and red groups: (a) Use an existing piece to design a new part or make measurements; (b) Cut a piece using a building block reference or cut with the help of a ruler; (c) Manipulate cut and assembled pieces to explore dimensions and symmetries, or perform stability tests.

Design Workshop with Novices

To better understand the challenges of physical prototyping by non-experts, we ran a workshop with 11 *novice* makers (6 women) including two post-docs, seven Ph.D. students, an intern, and a professional graphic designer. Because the study took place in early December, the design brief was to create an innovative *gingerbread "advent calendar"house*, with the winner to be constructed out of gingerbread.

We divided participants into four competing groups (blue, red, yellow, and green). Collaborators could use pen and paper to iterate on their early design ideas but were required to build their prototypes from 8 color-coded foamcore sheets of size DIN A4. We provided scissors and cutters for cutting, tape and glue for connecting multiple components, and rulers for measurements. The workshop lasted approximately four hours and was videotaped by three members of the research team.

Results. All four groups successfully fabricated a foamcore prototype gingerbread house. Each group generated a unique idea and faced a range of different construction challenges. Even so, we observed several common strategies: (1) using physical parts as templates for measuring or designing other parts of the construction; (2) cutting pieces with the help of pencil annotations and rulers, or using other construction blocks as reference; and (3) moving individual pieces around to explore alternative solutions, such as design symmetries, and combining blocks to test the stability of their constructions (see Figure 2).

Participants reported that physical prototypes were especially useful for crystallizing their ideas and solving construction problems. For example, the red team's concept consisted of multiple modular geometric shapes. They transitioned naturally from sketching to prototyping, and decided on the modules' geometry by first cutting and folding their paper sketches, and then moving to foamcore. However, participants also identified several shortcoming of this fully manual process, including the cumbersome nature of cutting multiple identical pieces, working with symmetries, correcting mistakes, making rapid measurements, rescaling individual pieces or blocks, and iterating on them further. These results suggest that *ShapeMe* should concentrate on easing the transition between rough

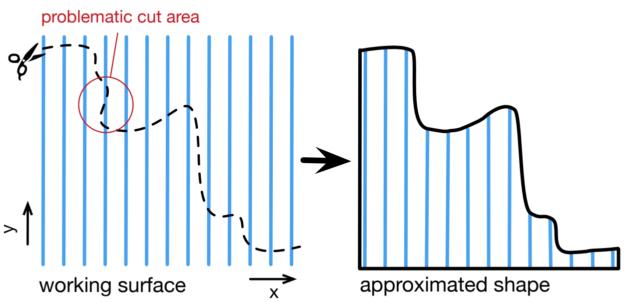

Figure 3. The *ShapeMe* approach for approximating 2D shape uses a grid of length-aware sensors. The red circle highlights the part of the cutting path that is not accurately captured by this sensing topology.

physical prototypes and digital models. We base our approach on a new sensing technology for producing shape-aware physical models.

APPROXIMATING SHAPE: APPROACH & CHALLENGES

Streaming the changing geometry of physical models while an artist work with her own tools is a challenge for current systems. Camera-based approaches require a setup of external cameras and usually suffer from occlusion problems. Smart tools [44], on the other hand, assume that the physical model remains fixed, or that additional tracking and calibration mechanisms are required. Moreover, artists cannot use their own sets of tools to reshape an object. We address these limitations by integrating the shape sensors into the actual material.

Material-based sensing technologies cannot currently support 2D or 3D shape sensing. Unfortunately, developing cost-effective technologies that can precisely sense arbitrary geometries is infeasible or at least hard for the moment. Thus, we focus on solutions that can provide an approximation of a shape through 2D and 3D sampling.

An overview of our approach is presented in Figure 3. We approximate shape by using multiple sensors that take the form of thin lines. The sensors are positioned in a two-dimensional space and are aware of their length. Since we also know their position in space, we can approximate a 2D shape. In the example of Figure 3, sensors are placed in parallel along the x axis, and the surface is cut by following an arbitrary 2D curve. The sensors' positions provide the x coordinates of the curve's points, while their lengths provide their y coordinates.

This principle can be extended to 3D objects by stacking several layers of material-sensor pairs on top of each other (see Figure 4). Creating volumetric objects out of thinner sheets is a widely established method in model making, as we observed with the shoemaker, as well as in architecture [38].

This approach raises several challenges:

1. Available technologies do not support length sensing. We developed our own sensing technology from scratch.

2. The precision of shape approximation depends upon the density of the sensors. We produce hardware components for sensing grids that are sufficiently dense to support realistic design scenarios.

3. The shape and orientation of the sensors constrain which types of shapes can be detected. For example, the vertical zig-zag path highlighted in Figure 3 cannot be detected accurately by the specific topology of line sensors. We address this with a software toolkit that helps designers easily adapt available sensor topologies to their specific construction needs.

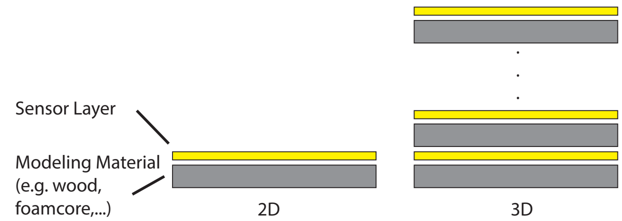

Figure 4. A *ShapeMe* sensor is placed on top of the sculpting material. Combining multiple layers on top of each other generates 3D objects.

FABRICATING SHAPEME MATERIAL

Our solution relies on the use of custom-made capacitors that can be embedded into common prototyping materials, such as paper and foamcore. We explain how we capture the varying length of a sensor through capacitance, how we make measurements, and finally, how we fabricate such sensors with off-the-self electronics.

Expressing Length through Capacitance

We implement length-aware sensors as parallel conductors that behave as parallel-plate capacitors. As shown in Figure 5, a parallel-plate capacitor consists of two conductive plates of length l and width w placed in parallel. When voltage V is applied, charges $+q$ and $-q$ appear on the surface of the two plates, and an electric field develops. Charges are proportional to the *capacitance* C of the capacitor. When the distance d between the plates is relatively small compared to their size, the capacitance of a parallel-plate capacitor is considered to be proportional to the surface area A of the plates [30]:

$$C = \varepsilon \frac{A}{d} \tag{1}$$

where ε is the permittivity of the dielectric medium that lies between the two plates.

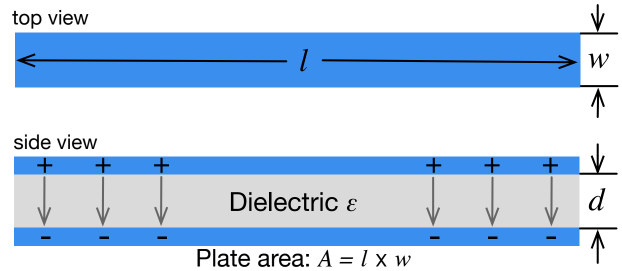

Figure 5. Schematic depiction of a parallel-plate capacitor.

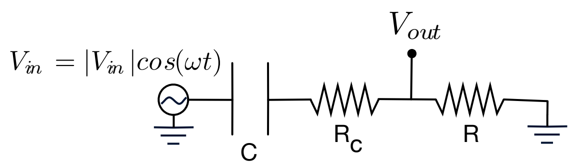

Figure 6. Electric circuit used to sense the length of the capacitor.

If we fix the width of the plates, their distance, and the dielectric medium, we can then express capacitance as a function of the length l of the plates such that $C(l) = c \cdot l$, where c is a constant that expresses capacitance per unit length. This linear relationship between the capacitance and the length of a capacitor leads to the core idea of our sensing technology. By placing two long flat conductors in parallel and in short distance along a dielectric, we create parallel-plate capacitors that know their length. If the capacitor is then cut by Δl, its capacitance will decrease by $\Delta C = c \cdot \Delta l$.

In practice, the relationship between the capacitance and the geometry of a capacitor is a more complex phenomenon, as the electric field extends beyond the overlapping area of the parallel plates. Yet, for long parallel plates of uniform width, in which were are interested here, the linear relationship between length and capacitance is not affected.

Length Estimation

To measure the capacitance of a sensor, we use a voltage divider, as the one shown in Figure 6. The voltage divider consists of the capacitor C and a resistor R. An imperfect capacitor may also have resistance. This resistance can be represented with a resistor R_c connected in series with the capacitor (see Figure 6). We can thus express the total resistance of the circuit as $R_{total} = R_c + R$. We can further write it as $R_{total} = \beta R$, where β represents the total resistance of the circuit as a percentage of R, thus $\beta \geq 1$.

Suppose we connect the circuit to an AC power supply of voltage V_{in} with an amplitude $|V_{in}|$ and an angular frequency ω. The ratio $r_\upsilon = |V_{out}|/|V_{in}|$ of the amplitude of the output voltage V_{out} to the amplitude of the source voltage V_{in} can be easily shown to be as follows:

$$r_\upsilon = \frac{\omega RC}{\sqrt{1+\beta^2(\omega RC)^2}} \tag{2}$$

From this, we can now derive the length of the capacitor as follows:

$$l = \frac{r_\upsilon}{\omega Rc\sqrt{1-\beta^2 r_\upsilon^2}} \tag{3}$$

where c is the capacitance per length unit. As explained earlier, this parameter can be considered as a known constant that depends on the permittivity ε and the thickness d of the dielectric medium, as well as the width w of the sensor.

If the resistor R is orders of magnitude larger than the capacitor's resistance ($R >> R_c$), then $\beta \simeq 1$. In this case, we can

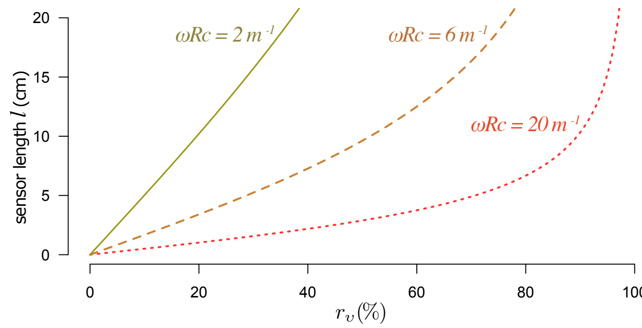

Figure 7. A sensor's length l as a function of the ratio $r_\upsilon = \frac{|V_{out}|}{|V_{in}|}$

directly estimate the length l of the sensor from the ratio r_υ of our voltage measurements. Figure 7 illustrates their relationship for three different ωRc configurations. Given a range of lengths of interest, we can choose an appropriate frequency and resistor to optimize the precision of our measurements. For our tests and prototypes, we target linear relationships, as this simplifies our approximation model and results in more accurate estimations of our sensors' changing length.

Printing the Sensors

To produce capacitors, we print two layers of conductive material in close distance, separated by an insulating layer that serves as the dielectric. We considered using double-coated sheets to directly print both layers of conductive ink with an inkjet printer [10], but unfortunately, double-coated sheets are not currently available on the market. An intermediate solution was to print the two conductive layers on separate sheets and then glue those together. However, this only works for rough prototypes because it is hard to precisely control the distance between two glued sheets. Our final solution was to to print the top layer of the sensors with an inkjet printer and use screen-printing [17] for the back layer. We used PEDOT:PSS as ink for screen-printing, but one can use other conductive inks with better conductive properties.

Our challenge was to deal with wiring scalability issues when the number of sensors increases. Each individual sensor requires two connecting points, one for each layer, but our solution illustrated in Figure 8 simplifies the problem. We use a common back electrode for a full grid of sensors that divides the number of required connectors by two. We investigated whether this solution influences the capacitance of the sensors, but our tests showed no clear effect. On the other hand, we

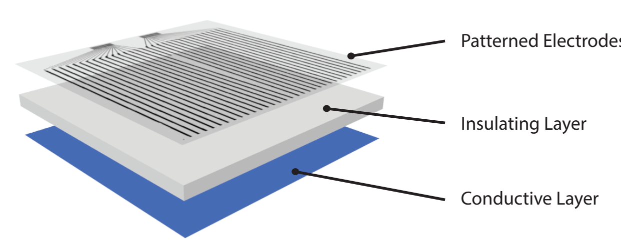

Figure 8. A *ShapeMe* grid of sensors consists of a layer of patterned electrodes (inkjet printed), a common back electrode (screen-printed), and an insulating layer between them.

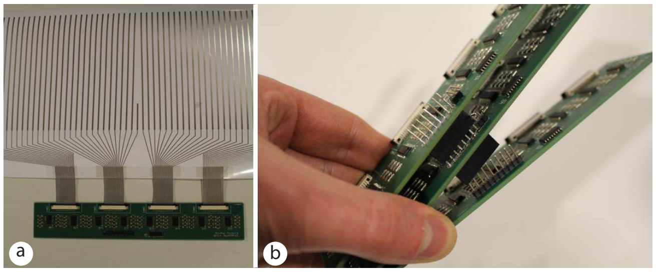

Figure 9. The *ShapeMe* Board. (a) Connected with an individual sheet of *ShapeMe* sensors. (b) For 3D models, we stack multiple boards together.

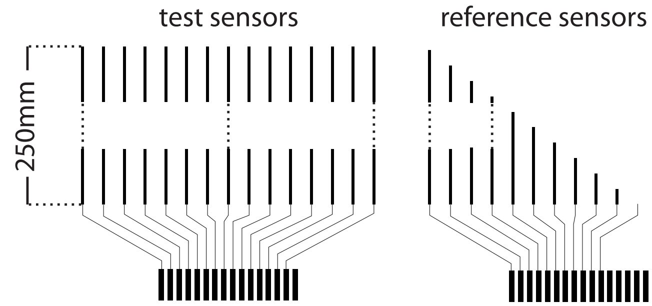

Figure 10. Configuration of sensors on our experimental sheets

observed an increase of the sensing values as more material was cut. This unexpected behavior was due to voltage and frequency changes in the AC signal of the commercial inverters that we tested. We address such problems by printing an additional sensor of constant length that serves as a "reference" for assessing how measurements are affected by the use of an imperfect power source. Our technical evaluation shows that the required adjustment of sensor values is linear. At each moment, we measure the ratio of the value change of the reference sensor and multiply to the measured value of the other sensors to correct their value.

We conducted additional tests to identify a minimum distance between sensors. We found that their values are influenced when distances become smaller than 1 mm. In particular, sensors become non-functional when their distance becomes 0.25 mm or lower. A minimum distance of 1.5 mm or greater can be considered as safe.

The ShapeMe Board

In our current implementation, a dense A4 sensor sheet can contain up to 64 individual sensors. Their total number further increases if we use multiple layers to capture a 3D shape. To deal with scalability issues, we implemented a modular *ShapeMe* Board (see Figure 9), a printed circuit board (PCB) that multiplexes 64 analog input (from the sensors) into one output channel. Multiple boards can be stacked one on top of each other to increase the number of readable sensors by 64 each at the cost of one more output channel. To connect sensors efficiently, we added four 16-pin FPC (flexible printed circuit) connectors to the front of the board. After experiments with several types of FPC connectors, we found that the type 1-84952-6 worked well with inkjet printed electronics. This implementation enables a standard Arduino Uno controller to process up to 384 sensors. We can support additional sensors by using a larger board, e.g., Arduino Mega, or by multiplexing through several *ShapeMe* boards in a recursive manner.

TECHNICAL EVALUATION

We conducted an experiment to assess the correctness of our model and evaluate the accuracy of our sensing technology.

Materials

We printed two sets of 11 *reference sensors* on two separate sheets of transparent foil (Mitsubishi NB-TP-3GU100). The width of the first set of sensors was 0.25 mm. The width of the second set of sensors was 1 mm. All reference sensors were shaped as straight lines, and their length ranged from 250 mm down to 0 mm, in steps of 25 mm. For each set of reference sensors, we also printed 15 *test sensors* on the same sheet of foil. All test sensors were initially 250 mm long and were shaped again as straight lines. Their width was identical to the width of the reference sensors (0.25 or 1 mm). Figure 10 illustrates the actual configuration of each sheet of sensors.

To print the front side of the sensors, we used an inkjet printer (Epson ET-2550) and Mitsubishi's inkjet printable silver nanoparticle ink (NBSIJ). We then screen printed a common back electrode of PEDOT:PSS (Gwent C2100629D1).

Procedure

Each set of 15 test sensors was cut simultaneously in 5 mm steps from 250 mm down to a 0 mm length. In order to ensure accuracy and repeatability, we used an Epilog Fusion M2 40 laser cutter to perform cuts. Reference sensors remained untouched during the full process.

After each cut, we captured 100 values per sensor by using the ShapeMe board. We used an inverter (WY-ELI) to supply the back electrode of the sensors with AC voltage: 72 V for the sensors of 0.25 mm and 37 V for the sensors of 1mm. We used a sine wave generator which generates signals at 1.6 kHz. This frequency allows sensing 320 sensors per second by averaging five measurements per sensor. This is sufficient for most realistic scenarios.

Results

We first analyzed the values of the 11 reference sensors at each step, from Length = 250 mm to 0 mm. To test the linear relationship between the true sensor length and the sensing values, we conduct simple linear regressions, for which we report their adjusted R^2 as a measure of goodness of fit.

We found perfect linear relationships for both sensor widths, where the worst goodness of fit was $R^2 = 99.7\%$ for $width = 0.25$ mm and $R^2 = 99.9\%$ for $width = 1$ mm. Nevertheless, the slope of the linear relationship did not keep constant. It progressively increased as further material was cut. As we discussed earlier, this is due to imperfect behavior of the inverter, whose voltage and frequency supply did not keep constant. Fortunately, we can measure this effect and remove it from our model. Given that the linear relationship is preserved, the ratio of voltage increase is overall common across sensors. Figure 11a shows how this ratio increases for the 11 reference

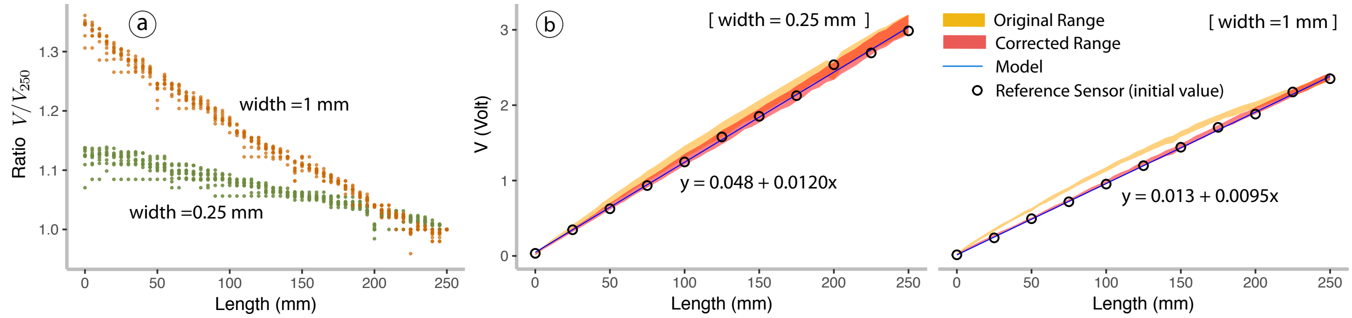

Figure 11. Results of Experiment 1: (a) Relative change of the 11 reference sensor values as the 15 active sensors are cut from Length = 250 mm to Length = 0 mm. (b) Range of values [min, max] of active sensors as their are progressively cut. In yellow, we show the range of the original values we measure, while in red, we show the corrected range by taking into account the relative change of a reference sensor. Circles show reference values.

sensors as the length of the active sensors is progressively reduced. We observe that on average, final voltage values increased by 12% (SD = 2%) for $width = 0.25$ mm and by 34% (SD = 2%) for $width = 1$ mm.[1]

We estimate this ratio from one or more reference sensors that remain uncut and apply it to the active sensors to correct their values. Figure 11b presents the range of values of the active sensor as they are cut. We show the original range of values that we measure in orange and the corrected range of values based on the change ratio of a reference sensor in red.[2]

After correction, sensor values fit well the linear model defined by the reference sensors within a error range that increases for longer sensors. We observe a tighter range of values for the wider (1 mm) sensors. However, the wider error for the 0.25 mm sensors is partly due to a single sensor whose values deviated from the rest of the sensors. Unfortunately, imperfections in the fabrication process can impact the accuracy of the measured values.

Table 1 reports the average length error (in mm) for two length estimation methods: (1) use a common linear function (see Figure 11b) for all sensors based on the values of the reference sensors; and (2) use a linear function that is specific to each individual sensor, where the function is derived by measuring the initial value of the original uncut sensor (per-sensor calibration). Mean errors are similar for both methods ($\sim$ 2.5 mm). However, per-sensor calibration results in smaller discrepancies between sensors.

Table 1. Average length error for two length estimation methods

	Average Sensor Length Error	
Width	Common Reference Model	Per-Sensor Calibration
0.25 mm	1.0 – 8.2 mm (M = 2.7 mm)	1.3 – 4.3 mm (M = 2.1 mm)
1 mm	0.7 – 4.7 mm (M = 2.1 mm)	1.8 – 2.9 mm (M = 2.4 mm)

DESIGNING SHAPEME MODELS

We envision the following typical design scenario: A maker, named Sally, starts with a rough *ShapeMe* model that serves as the initial construction material. This step requires: (1) choosing an initial shape, and (2) applying an appropriate sensing structure that can accurately capture cuts on the physical model. When the initial *ShapeMe* model is ready, Sally exports it to a PDF file that contains: (1) the sensing and wiring patterns to be printed with conductive ink using a inkjet printer on the front side of the sheets, (2) the back electrode pattern to be screen-printed (or printed with an inkjet printer if double-coated sheets are available), and (3) the outline of the initial shape to be cut with a laser cutter.

When the *ShapeMe* model has been prepared, Sally starts exploring its shape with manual cutting tools. When the model is attached to the *ShapeMe* board, its shape is captured and communicated to a digital modeling tool. Sally frequently switches to the digital model to explore symmetries, compare alternative variations or see the model in the context of a more complex 3D scene. At several steps, she reviews the history of past operations and returns to previous versions of the model by re-producing a physical model representation.

The ShapeMe Software Toolkit

We assist this process through a software toolkit that facilitates the design of *ShapeMe* material and enables its use in conjunction with digital 3D modeling tools. Figure 12 shows an overview of the toolkit's user interface as it communicates with the Unity platform.[3] The toolkit provides the following functions:

1. **Create an Initial Geometry.** Makers can choose and customize shapes from predefined or custom templates to define the initial geometry of a *ShapeMe* model. Each model consists of one or multiple layers formatted as printable A4 or A3 sheets. Each layer represents the 2D layer of a 3D structure (see Figure 4), or alternatively, an individual 2D surface, such as a wall or the ground of a house.

2. **Design a Sensing and Wiring Structure.** The user interface provides a collection of sensing structures from which users can choose. Makers can also customize the wiring structure that connects the sensors to the *ShapeMe* board by specifying layout constraints, e.g., minimum distances between wires, and selecting which connector groups to use. Makers can also customize the sensors' width and specify reference sensors to be used for value calibration (see Figure 12).

[1] The ACM version incorrectly reports values of 112% and 134%.

[2] Here, we use the longest reference sensor, but taking any other reference sensor (other than 25 mm for $width = 0.25$ mm) results in a similar range of values.

[3] https://unity3d.com/

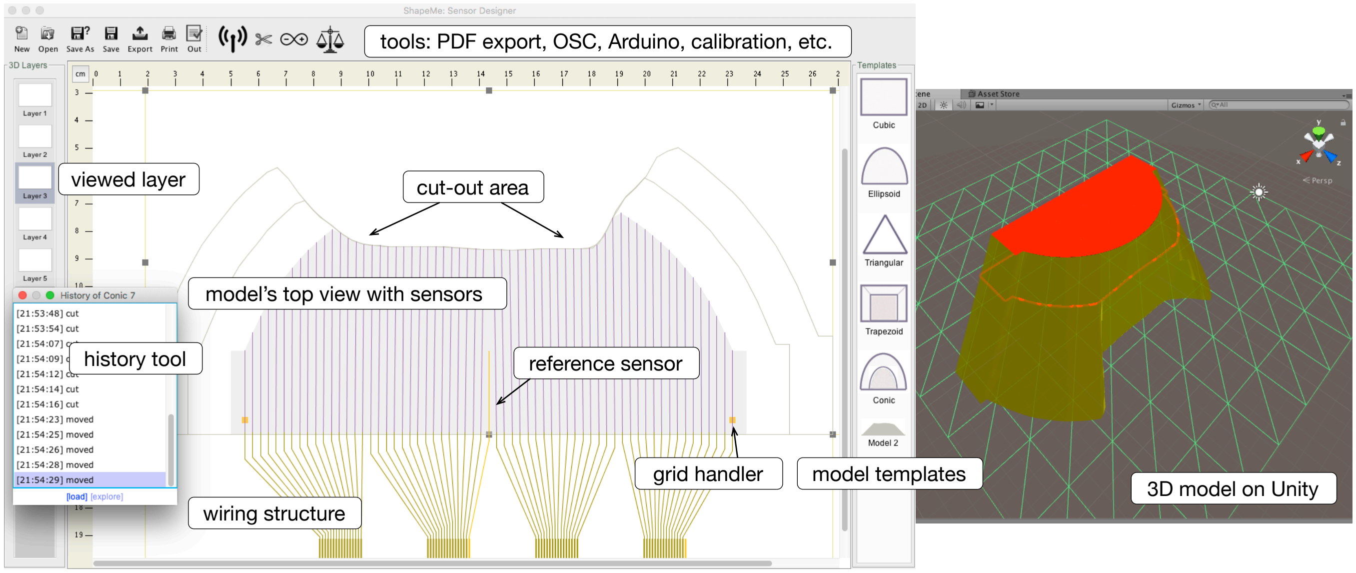

Figure 12. The *ShapeMe* toolkit (left) allows the maker to choose an initial geometry for the physical model and customize the topology of the length-aware sensors. The toolkit also applies a wiring structure to connect the sensors to the *ShapeMe* board. It then receives events from the hardware to update the shape of the virtual model. It keeps a history of the changes. It also communicates with Blender or Unity (right) through OSC messages, where the maker can view the 3D model reconstructed from its partial 2D layers.

3. **Export and Print a *ShapeMe* Model.** The tool can generate a PDF version of the model that describes both the shape of the model as well its sensing and wiring structure at different pages. Maker can use the PDF to print the sensors and wires through inkjet printing (or/and screen-printing) and initiate the shape of the model by laser-cutting.

4. **Detect Cut Events to Update the Model.** The tool communicates directly with the *ShapeMe* board through the USB port to collect raw sensing values. Based on these values, it estimates the updated length of the sensors. It then makes use of the sensing structure to update the 3D model. Makers can initially calibrate their sensors to improve sensing accuracy. We also provide a simulated cutting tool that allows the user to virtually cut the models and test the accuracy of different sensing structures.

5. **Communicate with 3D Modeling Environments.** The toolkit communicates with external digital modeling through the Open Sound Control (OSC) protocol [42]. We have implemented and tested communication with both Unity and Blender.[4] The first allows us to use the *ShapeMe* technology in conjunction with AR technologies, such as Microsoft Hololens, that support Unity. Bender, on the other hand, offers powerful modeling capabilities that many professional designers use. Our Blender plugin provides a range of modeling functions, such as easily creating symmetrical cut-out shapes. It further supports a bi-directional communication with the toolkit. Users can export model parts created on Blender to add grids of sensors and print their *ShapeMe* models to continue working on them with physical crafting tools.

6. **Review the Fabrication History and Iterate.** Makers can view the history of their edits and load past instances of a model. This allows them to *undo* [14] previous fabrication actions by applying a different sensing structure, re-printing the physical model, and following alternative fabrication paths. They can also save their sessions or create custom model templates for future re-use.

The implementation of the *ShapeMe* toolkit has been based on Java 8 and Java Swing. We use Text 7[5] to generate the printable version of the model, C# for the Unity plugin, and Python 3 for the Blender plugin.

Choosing an Appropriate Sensing Structure

The accuracy of shape detection is limited by how the model is cut with respect to its sensors' layout. Each sensor provides a single value, so it can only sense a single cutting point. As a result, the sensing capabilities of any sensor topology are limited to a certain family of shapes. For example, the vertical sensors in Figure 12 cannot accurately capture vertically oriented paths and completely fail to detect holes.

We explored a number of solutions to this problem, including sensors that take arbitrary shapes, e.g., curved lines, and branched sensor structures. However, our early experiments showed that capacitance values can be greatly affected by the shape of the sensors and their curviness. Unfortunately, complex sensor shapes require more sophisticated models that are harder to build. On the other hand, we observed that angular points have little effect on the sensed values, as long as their number is kept small. In particular, line-shaped sensors with one or two angular points seem to have a consistent behavior that is similar to the behavior of sensors with no corners and equal length. We thus decided to focus on sensing topologies with sensors consisting of one to three straight-line segments.

[4] https://www.blender.org/

[5] https://itextpdf.com

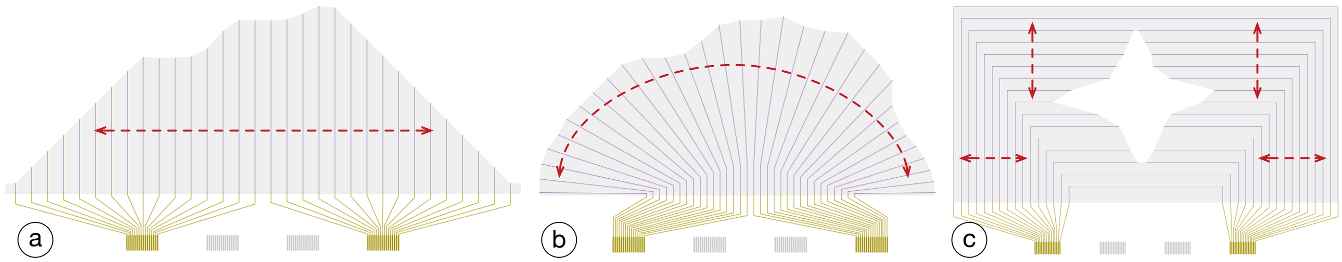

Figure 13. Alternative sensing topologies: (a) A layout of parallel vertical sensors, optimal for horizontal cuts. (b) A star layout, optimal for cuts around the periphery. (b) A topology of double-connected sensors, optimal for vertical cuts and holes. The red arrows show the direction of optimal cuts.

Figure 13 shows the three base sensor topologies supported by the *ShapeMe* toolkit for 32 sensors. Users can choose from these topologies and then customize their dimensions through UI controls (e.g., see grid handler in Figure 12). The first topology consists of vertical sensors and is optimal for horizontally oriented cuts. The second has a star layout and is optimal for sensing cuts around the periphery of the object. Finally, the third is best for sensing vertical cuts and is especially interesting as it also allows for detecting holes. This is achieved by using two connecting points for each sensor – if the sensor is cut, the *ShapeMe* board treats its pieces as two separate sensors that each provides a different value.

Makers do not have to stick to the layout of pins in the current implementation of *ShapeMe*'s board. Figure 14 shows an alternative configuration where connectors to the sensors can be attached at the center of the model. This configuration allows the maker to sculpture a model from any part of its periphery. Makers can also iterate on their models by applying a new sensing topology and reprint the model to start a new series of cuts. For example, Figure 14 shows how the user can change the sensors' layout to increase the sensing accuracy around a specific area of the model. Overall, our approach assumes that although makers do not precisely know the specific shape that they want to achieve, they have a rough idea about the parts of the model they need to work on and the type of their cuts. This assumption is consistent with other approaches [44] and our own observations of current professional practices.

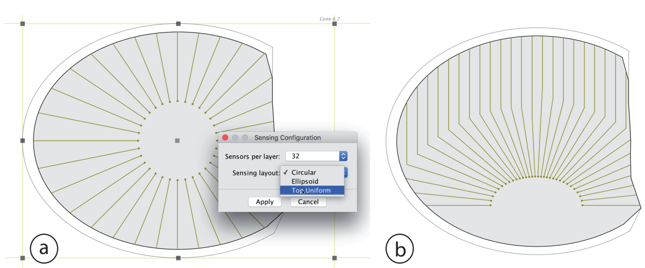

Figure 14. Sensor connections positioned at the center of the model. The user changes the structure of sensors from a star (a) to a star-to-parallel layout (b) to increase the precision of cuts on the top side of the model.

WALKTHROUGH SCENARIO

We demonstrate the use of the *ShapeMe* technology and software system through a construction walkthrough that is inspired by practices in architectural modeling. Our scenario has two parts. In the first part, a modeler creates the structure of a house. The workflow derives from the gingerbread-house prototyping scenario of our workshop with novice makers, where the physical model consists of multiple 2D planes. In the second part, another modeler works on the terrain. This part demonstrates how *ShapeMe* can work with 3D prototypes consisting of multiple layers stacked together.

Part 1: Modeling the House Structure

A modeler, Sally, decides to use a combination of physical prototyping tools and Blender to create the model of a house. She starts by designing one of its side walls. She uses the *ShapeMe* toolkit to create a 1 cm thick wall with a rectangular shape of 26 $\times$ 14 cm. The model of the wall is sent to Blender, where Sally copies it to create an identical wall on a parallel plane. She also creates the two perpendicular walls of the structure. Next, Sally decides to physically explore the shape of the first wall while watching how the updated shape affects the overall 3D structure of the house. She returns to the *ShapeMe* toolkit and prints a physical model of the wall. Because she now wants to cut horizontally, she selects a topology with vertically oriented sensors. She exports the model into a PDF file, which she uses to print the physical shape-aware wall model on foamcore, and then connects it to the *ShapeMe* board. She uses the *ShapeMe* toolkit to calibrate the sensor values. The shape of the foamcore model is then streamed to Blender, and shape changes are captured continuously by the toolkit's history tool.

Sally uses an ordinary cutter to make the first cut(see Figure 1a), and sees the the wall's new shape updated on Blender. Changes are also mirrored on the parallel wall, which helps Sally evaluate different shapes even though she is physically working on a single piece of the structure. When she is happy with a new shape, Sally returns to Blender, where she creates the house's roof, consisting of several rectangular parts (see Figure 1b). She decides to decorate part of the roof by manually crafting patterns on the downward side. She exports the roof from Blender to the *ShapeMe* toolkit and then creates the physical *ShapeMe* model, as before. When the physical model is printed, she cuts out a free-form pattern with scissors (see Figure 1c). At this point, Sally makes a mistake and cuts a piece that destroys her design. To correct it, she goes back in the history tool, loads the earlier version of the model (see Figure 15), and prints a new *ShapeMe* model. She proceeds, back and forth between physical and digital models, until she finalizes her design.

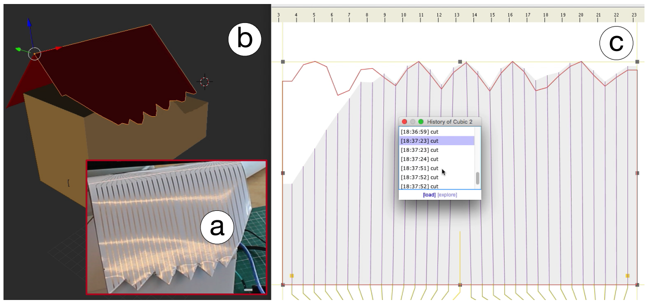

Figure 15. Correcting an accidental cut on the physical model (a, b) by going back to the history of cuts on the *ShapeMe* toolkit (c). The Blender user interface (b) shows the cuts on the symmetric piece of the roof.

Part 2: Modeling the Terrain

After the house is finished, Sally sends its digital model to Yeo, a befriended landscape modeler. Yeo likes working with 3D physical models as this lets him get a stereoscopic vision of a future landscape (see Figure 16a). He starts with a 3D model that consists of multiple foamcore layers. He uses the *ShapeMe* toolkit to create four shape-aware layers. As he wants to optimize the sensing precision around the periphery of each layer, he chooses a star-shaped sensing topology. After exporting and printing the *ShapeMe* sensors, he glues them to foamcore sheets and assembles the initial uncut physical model. He connect the model to the toolkit, which streams its geometry to the Blender plug-in.

After finishing his first prototype, Yeo decides to render the terrain together with Sally's house model. He notices that the top area is too small, so he scales up the model digitally (see Figure 16b). He then sends the finished digital model back to Sally, who exports the model and prints the complete physical model with a laser cutter.

LIMITATIONS AND FUTURE DIRECTIONS

The current version of *ShapeMe* has several limitations. The accuracy of our sensors is sensitive to their fabrication quality, while the assembly of multiple layers together is still a laborious process. A major constraint in our implementation is the lack of double-coated sheets that could be directly inkjet-printed on both sides. The use of screen-printing complicates the production of sensors and introduces additional sources of error. We investigated several solutions, but we expect our results to further improve in future iterations, e.g., by using double-coated sheets when they become available.

As makers use cutting tools to reshape a model, they may often get in contact with the *ShapeMe* sensors. We found that metallic tools can destabilize the sensing values, but the values return to their initial values quickly after the contact. Sensing values can be also affected by direct touch, thus we recommend protecting the sensors with insulation spray or by laminating the sheets.

Studies on design strategies [1] show that physical prototyping involves a range of tangible manipulations that extend beyond cutting, such as folding, rolling, and stretching. A future goal is to extend the *ShapeMe* approach to support flexible and stretchable substrates. Although the *ShapeMe* sensors are robust to bending, they cannot capture the level or the

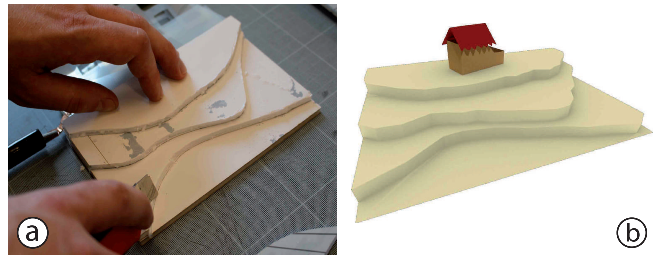

Figure 16. Working with a physical prototype that consists of four foamcore layers (a). The geometry of the model is continuously captured and can be rendered in Blender with the house model of part 1 (b).

position of a bending deformation. We believe that existing capacitive bend-sensing methods [6] could be integrated into our fabrication method, while strain sensing of silicon-based substrates could be based on techniques described by Cohen et al. [2] and Wessely et al. [39]. Future work needs to investigate how to best combine these approaches.

We are also interested in incorporating *ShapeMe* sensors into materials like foamcore and wood, e.g., by screen-printing layers of conductive and dielectric inks directly on their surface. Another challenging direction is how to develop sensor architectures that sense a wider range of physical geometries with a higher precision. Finally, our proposed modeling workflow and toolkit have not been evaluated with users. Future studies are required to assess whether and how they support the needs of either professional or novice makers.

CONCLUSIONS

Digital modeling tools offer features that are impossible with physical modeling, such as rescaling, undoing, and rapidly creating copies. However, digital models lack the subtle feel of physical materials and focus on creating a precise, final model, not exploring alternative shapes. To link these two worlds, we introduced *ShapeMe*, a novel smart material that captures its own geometry as it is physically cut by a maker. We presented a novel sensor technology that can detect 2D or 3D geometries by using grids of line-shaped capacitance sensors. Our technical evaluation showed that we can approximate the length of the sensors with a simple linear model. We also introduced a software toolkit that helps makers create their *ShapeMe* models, design their sensing layouts and wiring structures, export them for printing, and link them to 3D modeling software for live interaction. We ended with an application scenario that demonstrates the use of our technology and tools. Although our hardware implementation is still imperfect, we expect it to significantly evolve in future iterations.

ACKNOWLEDGMENTS

This research is funded by the European Research Council grant CREATIV (nº 321135).

REFERENCES

1. Adrien Bousseau, Theophanis Tsandilas, Lora Oehlberg, and Wendy E. Mackay. 2016. How Novices Sketch and Prototype Hand-Fabricated Objects. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 397–408. DOI:http://dx.doi.org/10.1145/2858036.2858159

2. Daniel J. Cohen, Debkishore Mitra, Kevin Peterson, and Michel M. Maharbiz. 2012. A Highly Elastic, Capacitive Strain Gauge Based on Percolating Nanotube Networks. *Nano Letters* 12, 4 (2012), 1821–1825. DOI: http://dx.doi.org/10.1021/nl204052z PMID: 22409332.
3. Artem Dementyev, Hsin-Liu (Cindy) Kao, Inrak Choi, Deborah Ajilo, Maggie Xu, Joseph A. Paradiso, Chris Schmandt, and Sean Follmer. 2016. Rovables: Miniature On-Body Robots As Mobile Wearables. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 111–120. DOI: http://dx.doi.org/10.1145/2984511.2984531
4. Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: Remixing Physical Objects with Copy and Paste from the Real World. In *Adjunct Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 381–382. DOI: http://dx.doi.org/10.1145/1866218.1866230
5. Sean Follmer, Daniel Leithinger, Alex Olwal, Nadia Cheng, and Hiroshi Ishii. 2012. Jamming User Interfaces: Programmable Particle Stiffness and Sensing for Malleable and Shape-changing Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 519–528. DOI: http://dx.doi.org/10.1145/2380116.2380181
6. Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. 2014. PrintSense: A Versatile Sensing Technique to Support Multimodal Flexible Surface Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1407–1410. DOI: http://dx.doi.org/10.1145/2556288.2557173
7. Daniel Groeger and Jürgen Steimle. 2018. ObjectSkin: Augmenting Everyday Objects with Hydroprinted Touch Sensors and Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 134 (Jan. 2018), 23 pages. DOI:http://dx.doi.org/10.1145/3161165
8. Felix Heibeck, Basheer Tome, Clark Della Silva, and Hiroshi Ishii. 2015. uniMorph: Fabricating Thin Film Composites for Shape-Changing Interfaces. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 233–242. DOI: http://dx.doi.org/10.1145/2807442.2807472
9. Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labune. 2012. Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials. *interactions* 19, 1 (Jan. 2012), 38–51. DOI: http://dx.doi.org/10.1145/2065327.2065337
10. Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. 2013. Instant Inkjet Circuits: Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices. In *UbiComp '13*. ACM International Conference on Ubiquitous Computing. http://research.microsoft.com/apps/pubs/default.aspx?id=201273
11. Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building Blocks for Swarm User Interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 97–109. DOI: http://dx.doi.org/10.1145/2984511.2984547
12. Danny Leen, Raf Ramakers, and Kris Luyten. 2017. StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 471–479. DOI: http://dx.doi.org/10.1145/3126594.3126643
13. Daniel Leithinger and Hiroshi Ishii. 2010. Relief: A Scalable Actuated Shape Display. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '10)*. ACM, New York, NY, USA, 221–222. DOI: http://dx.doi.org/10.1145/1709886.1709928
14. Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive Construction: Interactive Fabrication of Functional Mechanical Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 599–606. DOI: http://dx.doi.org/10.1145/2380116.2380191
15. Hyunjoo Oh, Tung D. Ta, Ryo Suzuki, Mark D. Gross, Yoshihiro Kawahara, and Lining Yao. 2018. PEP (3D Printed Electronic Papercrafts): An Integrated Approach for 3D Sculpting Paper-Based Electronic Devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 441, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3174015
16. Simon Olberding, Nan-Wei Gong, John Tiab, Joseph A. Paradiso, and Jürgen Steimle. 2013. A Cuttable Multi-touch Sensor. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 245–254. DOI: http://dx.doi.org/10.1145/2501988.2502048
17. Simon Olberding, Michael Wessely, and Jürgen Steimle. 2014. PrintScreen: Fabricating Highly Customizable Thin-film Touch-displays. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 281–290. DOI: http://dx.doi.org/10.1145/2642918.2647413

18. Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 579, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3174153

19. Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 887–896. DOI: http://dx.doi.org/10.1145/2858036.2858106

20. Huaishu Peng, Amit Zoran, and François V. Guimbretière. 2015. D-Coil: A Hands-on Approach to Digital 3D Models Design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1807–1815. DOI: http://dx.doi.org/10.1145/2702123.2702381

21. Cecil Piya, Vinayak Vinayak, Yunbo Zhang, and Karthik Ramani. 2016. RealFusion: An Interactive Workflow for Repurposing Real-World Objects towards Early-stage Creative Ideation. In *Proceedings of Graphics Interface 2016 (GI 2016)*. Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, 85–92. DOI: http://dx.doi.org/10.20380/GI2016.11

22. Ivan Poupyrev, Tatsushi Nashida, Shigeaki Maruyama, Jun Rekimoto, and Yasufumi Yamaji. 2004. Lumen: Interactive Visual and Shape Display for Calm Computing. In *ACM SIGGRAPH 2004 Emerging Technologies (SIGGRAPH '04)*. ACM, New York, NY, USA, 17–. DOI: http://dx.doi.org/10.1145/1186155.1186173

23. Majken K. Rasmussen, Esben W. Pedersen, Marianne G. Petersen, and Kasper Hornbæk. 2012. Shape-changing Interfaces: A Review of the Design Space and Open Research Questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 735–744. DOI: http://dx.doi.org/10.1145/2207676.2207781

24. Christian Rendl, Patrick Greindl, Michael Haller, Martin Zirkl, Barbara Stadlober, and Paul Hartmann. 2012. PyzoFlex: Printed Piezoelectric Pressure Sensing Foil. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 509–518. DOI: http://dx.doi.org/10.1145/2380116.2380180

25. Christian Rendl, David Kim, Sean Fanello, Patrick Parzer, Christoph Rhemann, Jonathan Taylor, Martin Zirkl, Gregor Scheipl, Thomas Rothländer, Michael Haller, and Shahram Izadi. 2014. FlexSense: A Transparent Self-sensing Deformable Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 129–138. DOI: http://dx.doi.org/10.1145/2642918.2647405

26. Alec Rivers, Andrew Adams, and Frédo Durand. 2012. Sculpting by Numbers. *ACM Trans. Graph.* 31, 6, Article 157 (Nov. 2012), 7 pages. DOI: http://dx.doi.org/10.1145/2366145.2366176

27. John Romanishin, Kyle Gilpin, and Daniela Rus. 2013. M-blocks: Momentum-driven, magnetic modular robots.. In *IROS*. IEEE, 4288–4295.

28. Anne Roudaut, Abhijit Karnik, Markus Löchtefeld, and Sriram Subramanian. 2013. Morphees: Toward High "Shape Resolution" in Self-actuated Flexible Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 593–602. DOI: http://dx.doi.org/10.1145/2470654.2470738

29. Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 103–108. DOI: http://dx.doi.org/10.1145/2807442.2807508

30. Paul Scherz and Simon Monk. 2016. *Practical Electronics for Inventors, Fourth Edition*. McGraw-Hill Education. https://books.google.fr/books?id=HZgXswEACAAJ

31. Hyunyoung Song, François Guimbretière, Chang Hu, and Hod Lipson. 2006. ModelCraft: Capturing Freehand Annotations and Edits on Physical 3D Models. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 13–22. DOI: http://dx.doi.org/10.1145/1166253.1166258

32. Ivan E. Sutherland. 1965. The Ultimate Display. In *Proceedings of the IFIP Congress*. 506–508.

33. Nirzaree Vadgama and Jürgen Steimle. 2017. Flexy: Shape-Customizable, Single-Layer, Inkjet Printable Patterns for 1D and 2D Flex Sensing. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction (TEI '17)*. ACM, New York, NY, USA, 153–162. DOI: http://dx.doi.org/10.1145/3024969.3024989

34. Ryoichi Watanabe, Yuichi Itoh, Masatsugu Asai, Yoshifumi Kitamura, Fumio Kishino, and Hideo Kikuchi. 2004. The Soul of ActiveCube: Implementing a Flexible, Multimodal, Three-dimensional Spatial Tangible Interface. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE '04)*. ACM, New York, NY, USA, 173–180. DOI: http://dx.doi.org/10.1145/1067343.1067364

35. Christian Weichel, Jason Alexander, Abhijit Karnik, and Hans Gellersen. 2015a. SPATA: Spatio-Tangible Tools for Fabrication-Aware Design. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*. ACM, New York, NY, USA, 189–196. DOI: http://dx.doi.org/10.1145/2677199.2680576

36. Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. 2015b. ReForm: Integrating Physical and Digital Design Through Bidirectional Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 93–102. DOI: http://dx.doi.org/10.1145/2807442.2807451

37. Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2991–3000. DOI: http://dx.doi.org/10.1145/2702123.2702391

38. M. Werner. 2011. *Model Making*. Princeton Architectural Press. https://books.google.fr/books?id=q0lTnQAACAAJ

39. Michael Wessely, Theophanis Tsandilas, and Wendy E. Mackay. 2016. Stretchis: Fabricating Highly Stretchable User Interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 697–704. DOI: http://dx.doi.org/10.1145/2984511.2984521

40. Amy Wibowo, Daisuke Sakamoto, Jun Mitani, and Takeo Igarashi. 2012. DressUp: A 3D Interface for Clothing Design with a Physical Mannequin. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, New York, NY, USA, 99–102. DOI: http://dx.doi.org/10.1145/2148131.2148153

41. Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2011. Interactive Fabrication: New Interfaces for Digital Fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 69–72. DOI: http://dx.doi.org/10.1145/1935701.1935716

42. Matthew Wright. 2005. Open Sound Control: An Enabling Technology for Musical Networking. *Org. Sound* 10, 3 (Dec. 2005), 193–200. DOI: http://dx.doi.org/10.1017/S1355771805000932

43. Ya-Ting Yue, Xiaolong Zhang, Yongliang Yang, Gang Ren, Yi-King Choi, and Wenping Wang. 2017. WireDraw: 3D Wire Sculpturing Guided with Mixed Reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3693–3704. DOI: http://dx.doi.org/10.1145/3025453.3025792

44. Amit Zoran and Joseph A. Paradiso. 2013. FreeD: A Freehand Digital Sculpting Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2613–2616. DOI: http://dx.doi.org/10.1145/2470654.2481361